Comunicación B-10

TÉCNICAS DE REDES NEURONALES APLICADAS AL RECONOCIMIENTO Y CLASIFICACIÓN DE *PIXELES* DE IMÁGENES DE SATÉLITE

Miguel Ángel Martínez Rubio
Servicio de Teledetección del INM

Mercedes Velázquez Pérez
Oficina de Proyectos del INM

RESUMEN

Las técnicas de redes neuronales son una de las técnicas objetivas más utilizadas en el reconocimiento automático de patrones y en la clasificación de datos de sensores remotos. Estos técnicas son útiles en el análisis no lineal de los procesos atmosféricos. En este articulo se mostrará elproceso de diseco, construcción, entrenamiento y aplicación de dos tipos de redes con diferente tipo de entrenamiento (supervisado y no supervisado) o una imagen procedente del satélite NOAA-TIROS. A cada «pixel» de la imagen se le asignará de forma automática un valor que representa: tierra, mar o la clase de nube a la que pertenece si es un «pixel» nuboso.

1. Introducción

Entre las razones de la amplia utilización de las redes neuronales, se pueden destacar las siguientes:

- Teóricamente pueden determinar cualquier función, por lo que son adecuadas en aplicaciones que no son fácilmente desciitas analíticamente.
- Excepto los patrones de entrada, no es necesario suministrar información adicional.
- Se pueden aplicar a cualquier tipo de patrones y a cualquier tipo de datos.
- Se obtienen buenos resultados con datos ruidosos, corno los encontrados frecuentemente en meteorología.

- No se hacen hipótesis acerca de la distribución estadística de las variables de entrada.

- Después de entrenadas son extremadamente rápidas y fácilmente implementables en arquitecturas paralelas.

Hay cientos de diferentes modelos de redes neuronales descritos en la literatura. Las redes difieren unas de otras en la topología y en el tipo de entsenamiento. De entre todos ellos hemos seleccionado dos redes: una con entrenamiento supervisado (descrita en el apartado 2) y otra con entrenamiento no supervisado (descrita en el apartado 3).

2. Entrenamiento supervisado (Perceptrón Multicapa)

En el proceso de entrenamiento de las redes de este tipo, se utilizan patrones conocidos y ya clasificados. Dentro de las redes con entrenamiento supervisado, la más utilizada es el Perceptrón Multicapa usando en el entrenamiento un algoritmo de retropropagación.

2.1. Topología

Consta de una capa de entrada, una o varias capas escondidas y una capa de salida. El número de neuronas que constituyen cada capa debe adaptarse a cada problema.

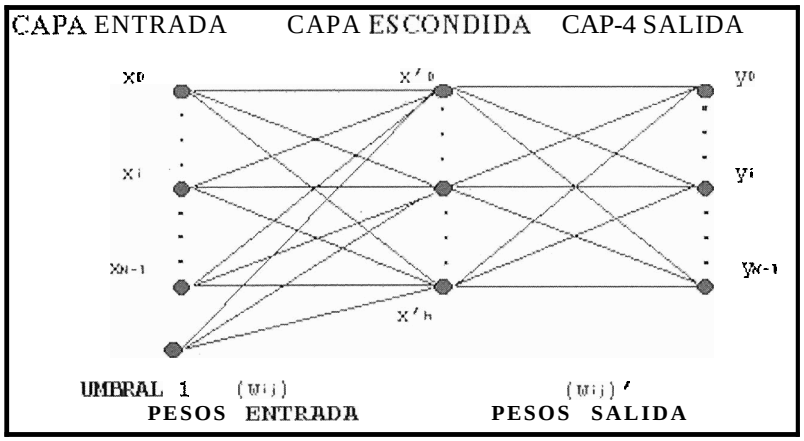

Fig. 1. *Diagrama de la topología del Perceptrón Multicapa*

Cada neurona de la capa escondida (x'_i) o de la capa de salida (y_l) tiene como entradas las neuronas de la capa anterior. El valor de salida de cada neurona esta representado por las siguientes funciones:

$$x'_i = f\left(\sum_{j=1}^{N-1} w_{ij} x_i - \theta_i\right) \qquad [1]$$

$$y_1 = f\left(\sum_{k=0}^{M-1} w'_{kl}\, x'_k - \theta'_k\right) \qquad [2]$$

donde w_{ij} y w'_{kl} son los pesos, que se determinarán en el proceso de entrenamiento y $f()$ es la función de activación.

2.2. · Función de activación

Como función de activación se emplean normalmente funciones continuas, crecientes, diferenciables y no lineales. La más utilizada es la función sigmoide binaria, debido a que la relación existente entre el valor de la función en un punto y su derivada, evita el cálculo de la derivada.

La función sigmoide binaria tiene como salida valores entre 0 y 1.

$$f(x) = \frac{1}{1+\exp(-\sigma x)} \qquad [3]$$

$$f'(x) = \sigma f(x)\,[1 - f(x)] \qquad [4]$$

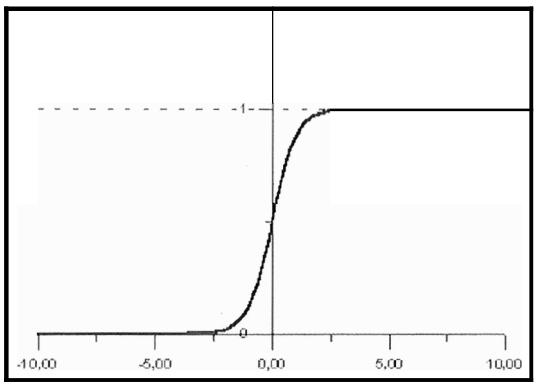

Fig. 2. *Función sigmoide binaria* con $\sigma = 1$

En algunos casos puede ser más adecuado utilizar la función sigmoide bipolar, cuyo rango de salida esta comprendido entre –1 y 1.

$$g(x) = 2f(x) - 1 = \frac{2}{1+\exp(-\sigma x)} - 1 \qquad [5]$$

$$g'(x) = \frac{\sigma}{2}\left[1+g(x)\right]\left[1-g(x)\right] \qquad [6]$$

2.3. Algoritmo de entrenamiento (BACKPROPAGATION)

El algoritmo de entrenamiento por retropropagación, es un algoritmo iterativo por descenso del gradiente diseñado para minimizar el error cuadrático inedio entre la salida real del Perceptrón Multicapa y la salida deseada. Consta de los siguientes pasos (asumireinos una función de activación sigmoide binaria que simplifica el cálculo de las derivadas):

Paso l. — Inicializar Pesos y Umbrales

Todos los pesos y umbrales de los nodos se inicializan con valores aleatorios pequeños.

Paso 2. — Presentar la entrada y las salidas deseadas

Se presenta un vector de entrada $(x_0, x_1, ..., x_{N-1})$ y la salida deseada $(d_0, d_1, ..., d_{M-1})$. Si la red se usa como un clasificador, todas las componentes de cada vector salida son 0 excepto la componente (o componentes) que corresponde al patrón de entrada que se fija a 1.

Paso 3. —Cálculo de las salidas reales

Usando las funciones sigmoides para cada neurona y a través de la topología de la red se calculan las salidas $(y_0, y_1, ..., y_{M-1})$

Paso 4. — Adaptación de los pesos

Se utiliza un algoritmo recursivo empezando en los nodos de salida y trabajando hacia atrás hasta llegar a la primera capa escondida.

Se ajustan los pesos mediante la siguiente fórmula (descenso del gradiente):

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j x'_i \qquad [7]$$

donde:

$w_{ij}(t)$: es el peso desde el nodo i-ésiino escondido en el instante t (o el peso desde una entrada) al nodo j-ésimo.

x'_i: es la salida del nodo i-ésiino (o es una entrada).

η: es un término de ganancia comprendido entre 0 y 1. Normalmente se usan valores pequeños, siendo conveniente hacer pruebas de convergencia y estabilidad con varios valores.

6,: es un término de error para el nodo j:

a) Si el nodo j es un nodo de salida, entonces:

$$\delta_j = y_j \cdot (1 - y_j) \cdot (d_j - y_j) \qquad [8]$$

donde d_j es la salida deseada del nodo j e y_j es la salida real obtenida por la red. En la fórmula [8] $y_j(1 - y_j)$ es la derivada de la función de activación, si se desea emplear otra función de activación sería necesario sustituir este factor por la derivada de la función de activación empleada. El uso de funciones como la sigmoide, en las cuales el valor de la función y su derivada están relacionados, permite acelerar el algoritmo de entrenamiento evitando hacer cálculos adicionales.

b) Si el nodo j es un nodo interno escondido, entonces:

$$\delta_j = x'_j \cdot (1 - x'_j) \sum_k \delta_k w_{jk} \qquad [9]$$

donde k es el índice que recorre todos los nodos de la capa por encima del nodo j.

Cuando hay problemas de convergencia es conveniente añadir un término de momento (efecto de memoria del cambio en el paso anterior) que evite que el proceso quede atrapado en algún mínimo local de la función de error. En este caso la actualización de los pesos se realiza mediante la fórmula:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x'_i + \alpha \left[w_{ij}(t) - w_{ij}(t-1) \right] \qquad [10]$$

donde $0 < \alpha < 1$.

Paso 5.— Repetir el proceso desde el paso 2 hasta que el error o los cambios en los pesos sean despreciables

2.4. Procedimiento operativo de entrenamiento del Perceptrón Multicapa

El esquema operativo para la construcción de la red, constará de las siguientes fases:

Fase l.— Selección de los patrones de entrenamiento

Se seleccionan un conjunto de patrones de entrada $(x_0, x_1, ..., x_{N-1})$ y sus correspondientes salidas deseadas $(d_0, d_1, ..., d_{M-1})$. Este conjunto de patrones disponibles se divide en un conjunto de entrenamiento y un conjunto de test.

Fase 2.— Aplicar iterativamente el algoritmo de entrenamiento («*BACKPROPAGATION*») a todos los patrones

Se aplica el algoritmo de entrenamiento a cada uno de los patrones del conjunto de entrenamiento. Una vez pasado el algoritmo a todos los patrones del conjunto de entrenamiento, se calcula el error total correspondiente a esa iteración pasa el conjunto de entrenamiento. El proceso de iteración se repite hasta que el error total de una iteración es menor que una cantidad prefijada, o hasta que éste se estabiliza. En este momento se salvan los pesos.

Con los pesos obtenidos en el paso anterior se aplica la red al conjunto de patrones que resewamos como conjunto de test. Si el error cometido al aplicar la red al conjunto de test es del orden del que obtuvimos al aplicar el algoritmo de entrenamiento al conjunto de patrones de entrenamiento, se considera que la red tiene un comportamiento adecuado. Si no, sería necesario repetir el proceso variando la topología, la función de activación, el algoritmo de entrenamiento o el tipo de red.

2.5. Aplicación a una imagen TIROS

La topología utilizada consta de una capa de entrada con 5 neuronas (brillo bandas 1 y 2, y temperatura bandas 3, 4 y 5), una capa escondida de 5 neuronas y una capa de salida con **6** neuronas (mar, tierra, nubosidad, nubes medias, nubes bajas y nubes de desarrollo).

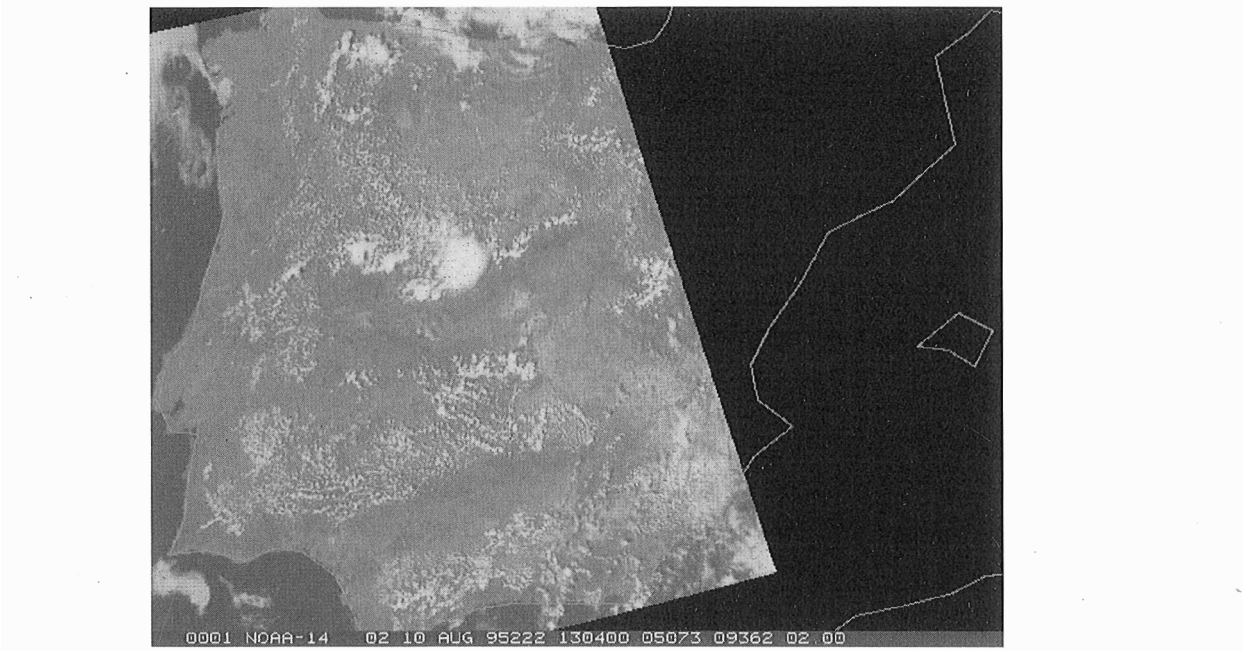

Fig. 3. *Imagen* ***del satélite*** *TIROS original*

El conjunto de entrenamiento se construyó capturando para cada «pixel» los valores de brillo de las bandas 1, 2 y de las temperaturas de las bandas 3, 4 y 5 sobre una zona pequeña (situada en la costa de Asturias que contenía todo tipo de patrones) de una imagen TIROS. Los valores de las correspondientes neuronas de salida se fijaron utilizando los valores asignados a la nubosidad por una máscara nubosa desarrollada en la Sección de Satélites, en el caso de que el «pixel» fuera despejado se le aporto información adicional sobre si ese punto era tierra o mar. Una vez entrenada la red se aplicó al área global con los resultados mostrados en la Fig. 4; cada una de las imágenes que constituyen dicha figura es el resultado de aplicar un realce a cada una de las neuronas de salida. Por ejemplo, en la Fig. 4.a) cuando la neurona de tierra despejada tiene un valor superior a 0,94 se le asigna el valor de brillo 250 y 0 en el resto de los casos.

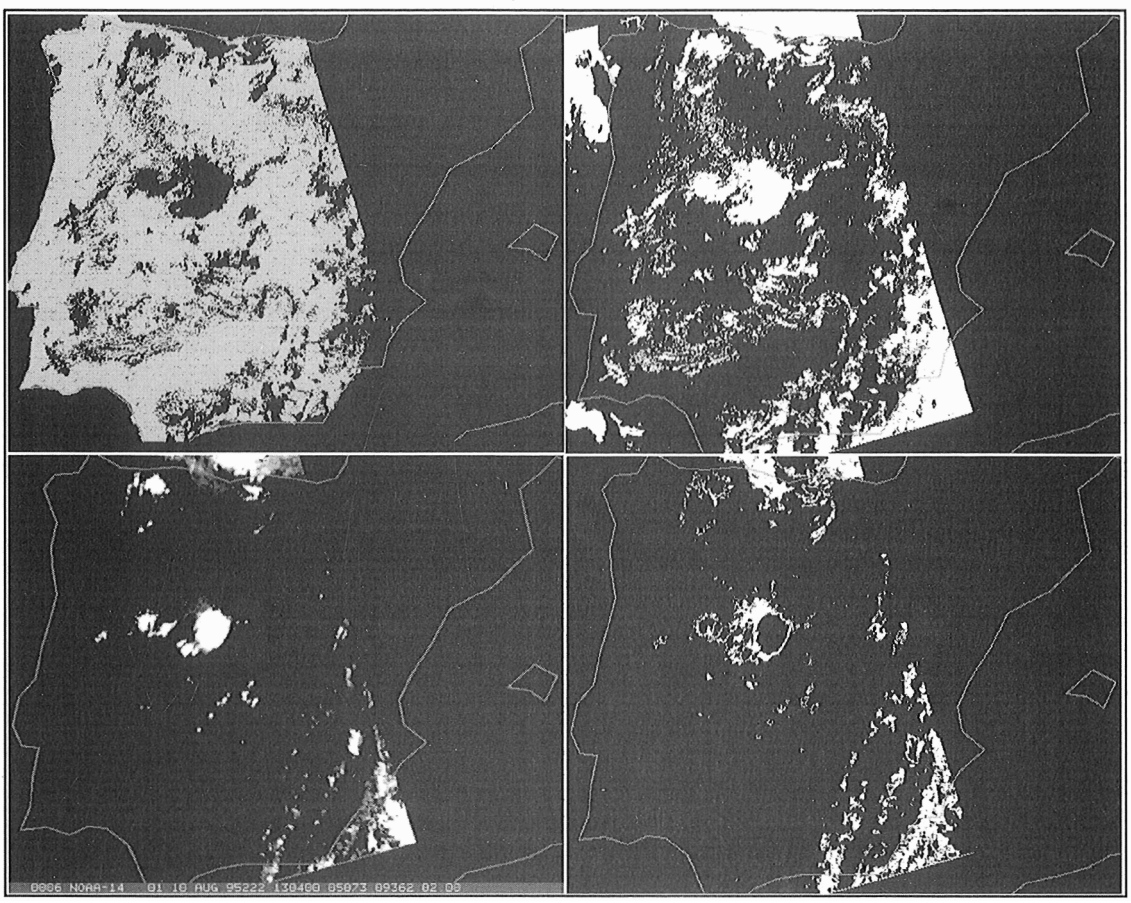

Fig. 4. *Imágenes resultantes de asignar valores de brillo a las diferentes neuronas.* a) *Tierra despejada, b) nubosidad,* c) *nubes de desarrollo,* d) *nubes medias*

3. Entrenamiento no supervisado. Mapas Topológicos Autoorganizativos (Mapas de Kohonen)

En este tipo de redes, en el proceso de entrenamiento no es necesario disponer de información «a priori» o tener clasificados los patrones del conjunto de entrenamiento, siendo el propio proceso de entrenamiento el que produce la clasificación.

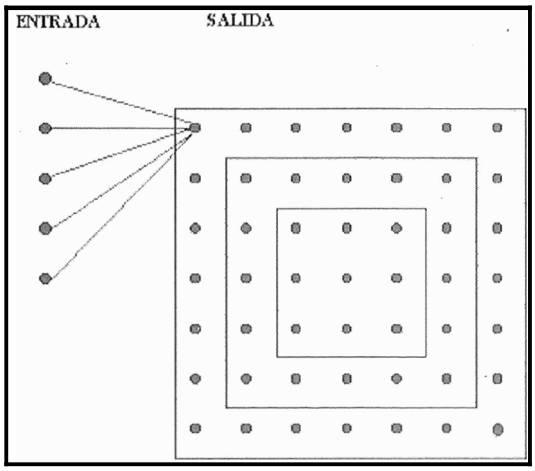

Fig. 5. *Topología de un Mapa de Kohonen*

3.1. Algoritmo para **producir** Mapas **Topológicos Autoorganizativos** (Mapas de Kohonen)

La Topología de los Mapas de Kohonen puede verse en la Fig. 5.

Consta de una capa de entrada y una capa de salida. En la capa de entrada se introducen los valores de los datos y se calcula la salida de cada neurona como producto escalar de los pesos de cada neurona por los valores de la capa de entrada. Después de entrenada, la salida para cada patrón es una matriz de distancias del patrón de entrada a los vectores peso de cada neurona. La disposición del mapa topológico como matriz permite fijar el criterio de vecindad, ya que las salidas (distancias) de

una neurona y las neuronas cercanas son similares. La interpretación de la matriz de salida puede ser todo lo compleja y exhaustiva que se desee, teniendo siempre en mente la interpretación geométrica de la red como conjunto de distancias del patrón de entrada a los vectores de peso de las neuronas; éstos tienen la propiedad de hacer mínimo el error total sobre el conjunto de entrenamiento. El algoritmo de entrenamiento es el siguiente:

Paso l. — Inicializar los pesos

Se inicializan los pesos desde las N entradas a los Mnodos de salida con valores aleatorios pequeños.

Se establece el radio inicial de vecindad.

Paso 2. — Presentar **nuevo** patrón de entrada

Paso 3. — Cálculo de la distancia a todos los **nodos**

Se calculan las distancias d_j entre la entrada y cada nodo ***j*** de salida usando

$$d_j = \sum_{i=0}^{N-1} \left[x_i(t) - w_{ij}(t) \right]^2 \qquad [11]$$

donde

$x_i(t)$ es la entrada al nodo i en el instante t y

$w_{ij}(t)$ es el peso desde el nodo de entrada i al nodo de salida j en el instante t.

Paso 4. — Seleccionar el nodo de salida con mínima distancia

Se selecciona el nodo j^* con 4 mínima.

Paso 5. — Actrializar los pesos del nodo j^* y sus vecinos

Se actualizan los' pesos para el nodo j^* y todos los nodos dentro del radio de vecindad definido por $NE_{j^*}(t)$ (disminuye al iterar).

Los nuevos pesos son:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t) \cdot [x_i(t) - w_{ij}(t)] \qquad [12]$$

para cada j E $NE_{j^*}(t)$ y para cada componente $0 \leq i \leq N-1$

Los términos $\eta(t)$ y $NE_{j^*}(t)$ disminuyen con el tiempo; $\eta(t)$ es un término de ganancia $(0 < \eta(t) < 1)$

Paso ***6:*** Repetir desde el paso **2,** hasta **que** $\eta(t)$ y $NE_{j^*}(t)$ sean próximos a cero

3.3. Aplicación a una imagen TIROS

Se escogió como conjunto de entrenamiento la imagen TIROS utilizada en el apartado 2.5 con resolución –10 (extraer un «pixel» de cada 10 de la imagen original); utilizando los valores de las cinco bandas de la imagen como componentes de los vectores de entrada. Se aplicó el algoritmo de la red de Kolionen al conjunto de entrenamiento y se salvaron los pesos resultantes (vectores de la red). La Fig. 6 ha sido generada asignando como valor de brillo a cada «pixel» de la imagen el índice del vector con distancia mínima, y aplicando posteriormente un realce de color que nos permita diferenciar los patrones.

Tanto este apartado como el 2.5 han sido iniplementados en entorno McIDAS.

Fig. 6. *Imagen TIROS clasificada mediante un mapa de Kohonen*

4. Conclusiones

Los resultados obtenidos indican que las redes neuronales son adecuadas para clasificar cada «pixel» de la iinagen, teniendo de esta forma una aplicación inmediata en teledetección. Además, estas técnicas son un posible camino para integrar información procedente de distintos sistemas de observación y obtener información adicional.

Las redes neuronales son generalizables a otro tipo de clasificaciones de patrones meteorológicos.

El hecho de usar funciones de activación no lineales nos permitirá aplicarlas en la descripción de procesos no lineales.

Su aplicación requiere un conocimiento adecuado del problema físico bajo estudio, para evitar que problemas asociados a este tipo de técnicas como pueden ser la existencia de mínimos locales, efectos de parálisis, etc.; nos impidan obtener los resultados óptimos.

Referencias

Fausset, L. Fundamentals of neural networks. Prentice-Hall, 1994.

Foody, G. M et al. The effect of training set size and composition on artificial neural network classification. International Journal of Remote Sensing, 16, 1707-1723.

Pankiewicz, G. S. A neural network cloud classifier trained with AVHRR data for use on Meteosat imagery. Proceedings of the 1995 meteorological satellite data users' conference, Winchester, pp. 393-400.

Stephanidis, C. N. et al. Tlie implementation of neural networks for cloud classification in digital satellite images. Proceedings of the 1995 meteorological satellite data users' conference, Winchester, pp. 435-441.

Agradecimientos

Agradecemos a Marcelino Manso lo sugerencia de aplicar las técnicas de redes neuronales a datos meteorológicos.